

KOREAN PATENT ABSTRACT (KR)

PUBLICATION

(51) IPC Code: G06F 15/16

(11) Publication No.: P2003-0022286

(43) Publication Date: 15 March 2003

(21) Application No.: 10-2003-7000705

(22) Application Date: 17 January 2003

(86) International Application No.: PCT/IB2002/01727

(86) International Application Date: 17 May 2002

(87) International Publication No.: WO 2002/95570

(87) International Publication Date: 28 November 2002

(71) Applicant:

KONINKLIJKE PHILIPS ELECTRONICS N.V.

(72) Inventor:

CHENG, Doreen, Y.

(54) Title of the Invention:

CODE GENERATION FOR INTEGRATING DEVICES INTO A MIDDLEWARE
FRAMEWORK

Abstract:

A middleware code generation system generates program code for integrating a device into a network. A variety of tools and a user interface are provided to create databases that contain the information required to generate the middleware integration code. The code generation system is configured to facilitate user modifications to the generated code, and particularly modifications to the code for error handling. In a preferred embodiment, the code generation system also generates a sample test application that serves to illustrate how to use the generated integration code in an application.

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.
G06F 15/16

(11) 공개번호
(43) 공개일자

특2003-0022286
2003년03월15일

(21) 출원번호 10-2003-7000705

(22) 출원일자 2003년01월17일

변역문 제출일자 2003년01월17일

(86) 국제출원번호 PCT/IB2002/01727

(86) 국제출원출원일자 2002년05월17일

(87) 국제공개번호 WO 2002/95570

(87) 국제공개일자 2002년11월28일

(81) 지정국 국내특허: 중국, 일본, 대한민국

EP 유럽특허: 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀란드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 터키

(30) 우선권주장 09/861,418 2001년05월18일 미국(US)

(71) 출원인 코닌클리크 필립스 일렉트로닉스 엔.브이.

네덜란드

네덜란드 엔엘-5621 베에이 아인드호펜 그로네보르세베그 1

(72) 발명자 청도린와이

네덜란드

네덜란드엔엘-5656에이에이아인드호펜홀스트란6

(74) 대리인 김창세

김원준

(77) 심사청구 없음

(54) 출원명 디바이스들 미들웨어에 통합시키는 프로그램 생성 시스템 및 생성 방법

명세서

기술분야

본 발명은 전자 네트워크 분야 및 가전에 관한 것이고, 상세하게는 디바이스들, 즉 자동 네트워크용 미들웨어 프레임워크와 같은 미들웨어 프레임워크에 통합시키는 프로그램 코드의 생성에 관한 것이다.

배경기술

DVD, D-VHS, 폰, 오디오 플레이어, 스틸 및 비디오 카메라 등과 같은 디지털 전자 제품은 점점 더 인기를 얻고 있고, 전형적으로는 네트워크화된 환경을 통한 이들 디바이스의 제어 및 관리를 용이하게 하도록 미들웨어를 규정하기 위한 컨소시엄이 형성되어 있다. HAVI, Jini 및 UPnP는 이러한 노력의 몇 가지 예이다. 일반적으로 공지된 바와 같이, "미들웨어"는 네트워크 상의 디바이스와 관련된 식별, 인증, 보안, 제어 등의 서비스를 제공하는 네트워크와 애플리케이션 사이의 소프트웨어의 레이어이다.

디바이스들 미들웨어 프레임워크에 통합시키기 위해 소프트웨어를 기록하는 것은 비단순(non-trivial) 태스크이다. 기초적인 기능의 소프트웨어에 더해서, 디지털 디바이스는 미들웨어가 요구하는 기능을 수행하고, 다른 디바이스 및/또는 소프트웨어가 네트워크를 통해서 디바이스의 성능을 액세스할 수 있고, 제어할 수 있게 하는 소프트웨어도 포함해야 한다.

다중 미들웨어의 존재는 문제를 더 악화시킨다. 디바이스의 판매자가 디바이스들 이러한 미들웨어에 통합시키기 위한 템플릿을 제공하려 한다면, 이 판매자는 적절한 템플릿이 사용되는 것을 보장하기 위해 적절한 방안은 물론, 다양한 템플릿을 제공해야 할 것이다. 이런식으로, 유저의 환경이 서로 다른 미들웨어를 구비한 다중 네트워크를 포함한다면, 유저는 이들 네트워크 각각의 프로그래밍 언어와 미들웨어의 요구조건을 학습해야 한다.

발명의 상세한 설명

본 발명의 목적은 네트워크에 디바이스들 인터페이스시키는데 필요한 프로그램 코드의 생성을 용이하게 하는 방법 및 시스템을 제공하는 것이다. 본 발명의 다른 목적은 네트워크 상의 디바이스의 제어를 용이하게 하는데 필요한 코드를 제공하는 것이다.

이러한 목적은, 디바이스들 네트워크에 통합시키기 위한 프로그램 코드를 생성하는 미들웨어 코드 생성 시스템을 제공함으로써 이루어진다. 미들웨어 통합 코드를 생성하는데 필요한 정보를 포함하는 다양한 룰과 유저 인터페이스가 제공된다. 바람직한 실시예에서, 코드 생성 시스템은 생성된 코드로의 유저의 수정을, 상세하게는 여러 핸들링을 위한 코드의 수정을 용이하게 하도록 구성된다. 바람직한 실시예에서, 코드 생성 시스템은 샘플 테스트 애플리케이션을 생성해서, 애플리케이션에서 생성된 통합 코드를 사용하는 방법을 나타내는 것을 돕는다. 더욱이, 바람직한 실시예에서, 코드 생성 시스템은 메이커파일(Makefile)과 같은 편집 도구(compilation facilities)를 생성한다.

도면의 간단한 설명

도 1은 본 발명에 따른 미들웨어 코드 생성 시스템의 블록도,

도 2는 본 발명에 따른 코드 생성 엔진의 블록도,

도 3은 본 발명에 따라, 디바이스의 HAVI 미들웨어로의 통합을 용이하게 하도록 FCM 코드를 생성하도록 코드 생성 시스템에 의한 사용을 위해, HAVI 명세 데이터의 기능 제어 모듈(Function Control Module:FCM) 구성요소에 유지되는 정보의 블록도,

도 4는 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해, FCM 구성요소의 데이터 구조 구성요소에 유지되는 정보의 블록도,

도 5는 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해, FCM 구성요소의 서비스 애플리케이션 프로그램 인터페이스(API) 구성요소에 유지되는 정보의 블록도,

도 6은 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해, FCM 구성요소의 이벤트 구성요소에 유지되는 정보의 블록도,

도 7은 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해, FCM 구성요소의 통지 속성(notification attribute) 구성요소에 유지되는 정보의 블록도,

도 8은 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해, 개인용 링크(Private Link) 데이터 베이스의 개인용 링크 구성요소의 통지 속성 구성요소에 유지되는 정보의 블록도,

도 9는 본 발명에 따라, 미들웨어 코드 생성 시스템에 의한 사용을 위해 특정 링크 타입의 링크 상태 응답 구성요소에 유지되는 정보의 블록도,

도 10은 본 발명에 따라, 미들웨어 코드 생성 시스템에 생성된 코드의 설치과정의 블록도.

도면에 있어서, 동일한 참조 번호는 유사한 대응하는 특성 또는 기능을 나타낸다.

실시예

도 1은 본 발명에 따른 미들웨어 코드 생성 시스템(100)의 블록도의 실시예를 도시하고 있다. 코드 생성 시스템(100)은 소프트웨어 툴(110), 데이터 베이스(120, 125, 130, 135) 및 디바이스 미들웨어 및 통합 코드를 포함하고 있는 코드 생성 엔진(150)을 포함한다. 전자 디바이스를 HAVI 네트워크와 같은 미들웨어 프레임워크에 통합시키는 소프트웨어의 자동 생성을 가능하게 하도록 구성된다. 소프트웨어 툴(110)은 통합 코드(180)의 모듈을 생성하기 위해 코드 생성 엔진(150)에 의해 요구되는 정보(120-130)를 제공하기 위해 사용되는 유저 인터페이스를 제공한다. 인터페이스 디바이스(110)가 사용되어서 코드 생성 프로세스를 제어하고, 생성된 코드를 생성하고 업데이트한다. 정보(120-135)는 데이터 베이스의 조합이 조직되어 있는 바와 같이 도시되어 있고, 여기서 용어 데이터 베이스는 코드 생성 목적으로 구성된 정보를 포함하는 논리 데이터 저장부를 의미하기 위해 사용된다. 정보(120-135)는 특정 데이터 베이스(120), 스태틱 정보 데이터 베이스(125), 템플릿 데이터 베이스(130) 및 개인용 링크 데이터 베이스(135)를 포함한다. 트루 데이터 베이스가 사용될 수도 있지만, 인코어(In-core) 데이터 구조 및 파일도 사용될 수 있다.

동작시, 시스템(100)의 유저는 코드(180)가 생성되는 디바이스의 특성 및 기능에 대해서 알고 있다. 전형적인 코드 생성 과정에서, 유저는 소프트웨어 툴(110)을 사용해서 코드(180)가 생성되는 디바이스의 타입 및 디바이스와 통신에 사용되는 개인용 링크의 타입을 나타낸다. 예컨대, 유저는 디바이스가 VCR이고, 서브 유닛으로서 튜너, 테이프 레코더 및 테이프 플레이어 포함하고 있다는 것을 알릴 수 있다. 소프트웨어 툴(110)을 사용해서 유저는 미들웨어 명세에 의해 커버되지 않은 디바이스 특성을 지정할 수 있다. 바람직한 실시예에서, 코드 생성 시스템(150)은 통합 코드 모듈(180) 및 디바이스용 간단한 테스트 애플리케이션(185)을 생성한다. 통합 코드(180)가 제공되어서 디바이스를 미들웨어 환경에 통합시키고, 테스트 애플리케이션(185)이 제공되어서 애플리케이션에서 통합 코드(180)를 사용하는 방법을 나타낸다. 코드 통합 엔진(150)은 생성된 코드(180)에서 유저 변형이 유저의 수정방안이 의뢰된 부분에 인덱스를 제공하는, 라인 번호와 파일 이름과 같은, 포인터의 리스트를 생성한다. 수정 정보는 바람직하게는 유저의 수정 방안(145)을 위한 가이드의 형태로 제공되고, 이들 수정 방안은 유저가 미들웨어의 인식을 필요로 하지 않도록 제공되는 것이 바람직하다. 전형적으로, 제안된 수정 방안(145)은 여러 핸들링 코드를 포함하고 있다. 코멘트가 코드(180) 내에 위치되어서 코드(180)를 수정하는 방법 및 이유를 유저에게 제공한다. 바람직한 실시예에서, 소프트웨어 툴(180)은 코드(180)의 코드 세그먼트를 유저에게 하나씩 나타내어서, 전체 처리 과정에서 유저를 안내하는 인터페이스를 제공한다. 바람직한 실시예에서, 코드 생성 시스템(100)은 생성된 코드용 메이크 파일과 같은 편집 도구를 생성하도록 구성되기도 한다.

코드 생성 과정은 원리적으로 미들웨어 명세에 의해 구동된다. 예컨대, HAVI 비디오 카세트 레코더(VCR) 기능 제어 모듈(FCM)의 코드 생성은 HAVI VCR FCM 명세에 의해 구동된다. 이를 수용하기 위해, 미들웨어 명세에서의 관련 정보는 코드 생성에 용이하게 사용될 수 있는 형태(120)로 변환된다. 소프트웨어 툴(110)은 명세를 명세 데이터 베이스로 변환하는 것을 돕는다. 이들 툴(110)을 사용해서, 두 단계로 변환이 이루어진다.

단계 1: 유저 가이드형 자동 변환. 소프트웨어 툴(110)은 명세 텍스트의 구문에 기초해서 명세의 정보를 명세 데이터 베이스로 변환하는 파서(parsers) 및 포매터를 포함한다. 이 단계에서는 필요한 정보의 주요부분을 생성한다. 이 툴을 사용해서 유저는 명세에서 그들의 업무와 관련된 부분만을 선택할 수 있다. 예컨대, 회사가 VCR 및 카메라만을 생산한다면, 명세 중에서 VCR 및 카메라에 관련된 부분의 정보만이 사용된다.

단계 2: 유저 증가 및 명세. 의미 정보가 예컨대 UPnP의 경우와 같이, 명세에 정식으로 규정되면, 정보가 추출되어서 특정 데이터 베이스(120)에 유지된다. 그러나, 만약 HAVI의 경우와 같이 의미 정보가 명세에 정식으로 표현되어 있지 않으면, 소프트웨어 툴(110)을 사용해서 유저는 명세에 의해 규정된 의미에 기초해서 명세 데이터 베이스(120)를 증가시킨다. 의미 정보는 전형적으로 디바이스와 관련된 지식 및 디바이스용 명세의 이해로부터 획득된다. 예컨대, 플레이 커맨드는 VCR의 전송 모드를 밸류 플레이로 변환하고, 이 변환은 이벤트를 트리거한다. 유저는 툴을 사용해서 미들웨어 명세 및 개인 링크 정보에 의해 커버되지 않는 특성을 명시한다. 예컨대, 카메라가 GPS 성능을 가지고 있다면, 이 성능은 적절한 코드의 생성을 위해 명세 데이터 베이스에 추가되어서, 미들웨어에 이 성능을 주입할 필요가 있다. HAVI의 경우에, 이 성능은 카메라 디바이스의 FCM에 따라서 모델링될 수 있다. 유저는 툴을 사용해서, 미래의 코드 구성에 영향을 미치도록 템플릿(130)을 수정할 수 있다.

하기의 설명에서, 용어 모듈 및 메소드(기능)는 프로그래밍 언어에 규정된 표준 규정을 따른다. 종래의 기술에 금지된 바와 같이, 자바와 같은 객체 지향 언어는 "메소드"를 사용하고, C, 파스칼 및 포트란과 같은 절차언어는 "절차", "서브루틴" 및 "기능"을 사용하는 등, 다른 프로그래밍 언어는 연산 그룹에 대해서 다른 기술을 사용한다. 이해를 돕기 위해서, 용어 "메소드(기능)"는 "메소드"의 프로그래밍 해석을 구별하기 위해 사용된다. 이 코드 세그먼트는 일 이상의 연속적인 프로그램 명령문의 순서를 나타내기 위해 이 애플리케이션에서 사용된다.

타겟 코드(180)의 아키텍처 및 코딩 스타일(생성될 코드)은 코드 생성 알고리즘을 간략화하도록 설계된다. 타겟 코드 모듈(180)은 소수의 그룹으로 분류될 수 있고, 그 그룹의 모듈의 구조가 유사한 방식으로 따르도록 설계된다. 판매자의 이름, 디바이스의 모델 및 타입과 같은 파라미터가 사용되어서 무관한 모듈로부터 소망의 코드 모듈을 구별한다. 예컨대, 유저는 "필립스사에 의해 제조된 VCR 모델 xyz"를 사용해서 생성된 코드 모듈을 명시한다. 사용된 코드의 타입(예컨대, FCM 또는 DCM) 및 프로토콜의 타입과 같은 파라미터가 사용되어서, 소망의 모듈을 더 선택하기 위해 사용될 수 있다. 예컨대, 유저는 개인 링크와 같이 1394에 대한 AV/C 프로토콜을 사용하는 VCD용 FCM 생성기를 선택할 수 있다. 유사하게, 메소드(기능) 및 모듈의 코드 세그먼트는 그룹으로 분할될 수 있고, 그룹내의 메소드(기능) 및 세그먼트는 각각 유사한 패턴을 따른다. 후술되는 바와 같이, 전술한 타겟 코드(180)의 디자인은, 타겟 코드(180)의 아키텍처를 인코딩하기 위해, 코드 생성 엔진(150)에서의 코드 생성기(252, 254, 256:도 2) 및 템플릿(130)의 사용을 가능하게 한다. 바람직하게는, 유사한 코드 모듈의 그룹의 골간(skeleton)은 모듈 생성기(254)에서 캡슐화된다. 모듈 생성기(254)는 코드 모듈(284)을 생성한다. Client_Module_Generator 및 Server_Module_Generator가 모듈 생성기의 예이고, 자바 클래스 모듈은 코드 모듈의 예이다. 유사한 메소드(기능)의 그룹의 골간은 메소드 생성기(252)에서 캡슐화된다. 메소드 생성기(252)는 메소드(282)를 생성한다. Client_Asynchronous_Method_Generator 및 Server_Operation_Method_Generator는 메소드 생성기의 예이다. 유사한 코드 세그먼트의 그룹의 골간은 코드 세그먼트 생성기(256)에서 캡슐화된다. 코드 세그먼트의 예는 인클루드 세그먼트, 여러 핸들링 세그먼트 등을 포함한 경고 세그먼트이다. 도 2의 각각의 템플릿(210, 220), 즉 메소드 템플릿(210) 및 코드 세그먼트 템플릿(220)은 차이를 태그한다. 코드 세그먼트 생성기(256)는 차이풀 리졸브(resolve)해서, 연속 문장(statement:286)의 순서를 생성한다. Argument_List_Generator 및 Include_Block_Generator는 세그먼트 생성기의 예이다.

문장 그룹의 패턴 및 간단한 메소드(기능)의 그룹의 패턴은 템플릿(130)에서 캡슐화된다. 템플릿(130)은 "C"와 같은 선택된 프로그래밍 언어에 따라서 코드 세그먼트에서 고정된 부분과 가변 부분을 포함한다. 템플릿의 가변 부분은 리졸브되어서 템플릿(130)을 코드(180)의 유효 코드 세그먼트로 변환시킨다. 코드 세그먼트 생성기는 일 이상의 템플릿을 포함하는 것이 전형적이다. 메소드(기능)의 코드가 템플릿 변수를 리졸빙함으로써 생성될 수 있다면, 템플릿이 간단한 메소드(기능)을 나타내도록 사용될 수 있다.

하기 설명에서, 용어 트리, 서브 트리, 노드, 리프(leaf), 패턴트 및 차일드는 전산학에서 사용되는 표준 규정을 따른다. 바람직하게는, 코드 생성 엔진(150)을 오가면서 코드를 생성하도록 구성된다. 전형적으로 트리의 루트는 FCM 서버 프로그램 또는 FCM 클라이언트 프로그램과 같은 독립형 프로그램용으로 생성된 모든 모듈을 나타낸다. 루트의 칠드런은 프로그램에 의해 요구된 코드 모듈용 모듈 생성기이다. 모듈 생성기 노드는 모듈의 구성요소를 생성하는 서브 트리의 루트이다. 서브 트리의 노드는 메소드 생성기 및 코드 세그먼트 생성기이다. 유사하게, 메소드 생성기를 나타내는 노드는 메소드의 구조를 캡처하는 서브 트리의 루트이다. 서브 트리의 노드는 다른 메소드 생성기 및 코드 세그먼트 생성기이다. 프로그램 트리의 리프는 코드 세그먼트 생성기를 포함한다. 코드 생성 과정의 종료를 보장하기 위해, 서브 트리의 루트는 동일한 서브 트리의 어떤 노드로서는 나타나지 않는다. 예컨대, 모듈 생성기는 자신의 서브 트리의 비 루트 노드로서는 나타나지 않고, 메소드 생성기는 자신의 서브 트리의 내부 노드로서는 나타나지 않는다.

전술한 바와 같이, 타겟 코드(180)는 코드 생성기(252, 254, 456)에 의해 생성된다. 모든 코드 생성기는 예컨대, 모듈, 기능 또는 기능 세그먼트와 같은 타겟 코드(180)의 특정부분을 생성하는 메소드에 관한 지식을 가지고 있다. 코드 생성 과정 동안, 생성기(252, 254, 256)는 디바이스의 타입, 모델 및 판매자와 같은 유저 입력을 사용해서, 데이터 베이스(120, 125, 135)로부터 정보를 찾고, 이 정보를 사용해서 의무적으로 코드의 일부를 생성한다. 생성기가 템플릿(130)을 만나면, 데이터 베이스(120, 125, 135)로부터의 지식을 사용해서, 템플릿(130)의 가변 부분을 분석하고, 템플릿(130)을 소망의 코드로 변환시킨다. 트리를 오가는 동안, 패턴트 코드 생성기가 차일드 코드 생성기를 만나면, 패턴트 코드 생성기는 자신의 코드 생성을 계속하기 전에 차일드 생성기를 호출해서 코드의 대응하는 부분을 생성한다. 예컨대, 모듈 생성기가 메소드 생성기를 만나면, 메소드 생성기를 호출해서 소망의 메소드를 생성한다. 메소드가 생성된 이후에, 패턴트 모듈 생성기는 자신의 일을 재개한다. 유사하게, 메소드 생성기가 차일드 메소드 생성기 또는 차일드 코드 세그먼트 생성기를 만나면, 차일드 코드 생성기를 호출하고, 차일드 생성기가 종료된 이후에만 자신의 일을 재개한다.

도 1에 도시된 바와 같이, 본 발명의 바람직한 실시예에서 코드 캐시(155)를 사용해서 유저는 생성된 모듈 및 미래의 코드 생성용 메소드(기능)를 저장할 수 있다. 캐시(155)는 코드에 대한 포인터(파일명과 같은) 및 코드가 생성된 조건을 포함한다. 이러한 조건은 모든 데이터 베이스의 버전 정보 및 전체 유저 인터페이스가 입력된 유저 명세를 포함한다. 코드 캐시(155)를 사용하는 전형적인 시나리오는 예컨대 새로운 특성을 추가하고, 코드의 재생성을 소망하는 것과 같이, 유저가 디바이스 명세(120)의 일부를 수정할 때이다. 본 발명의 바람직한 실시예에서, 코드 생성 엔진(150)은 자주 사용되는 부분에 대해서만 새로운 코드(180)를 재생성하고, 자주 사용되지 않는 부분에 대해서는 캐시(155)내의 코드를 사용하도록 구성된다.

도 1의 스택 정보 데이터 베이스(125)는 코드 생성에 필요한 미들웨어 스택 구현 정보를 포함한다. 데이터 베이스(125)는 또한 스택 구현의 버전의 트랙을 유지한다. 데이터 베이스(125)의 정보의 주요 부분은 스택에 선언된 유저 규정형 타입 및 선언을 포함하는 파일명이다. 이 정보는 인클루드 파일용 생성하기 위해 사용된다.

도 1의 템플릿 데이터 베이스(130)는 코드 생성 엔진(150)에 의해 사용되는 모든 템플릿을 포함한다. 데이터 베이스(130)의 바람직한 실시예는 정해진 디렉토리내의 파일로서 각각의 템플릿을 저장한다. 템플릿의 이름은 코드 생성에서 이 템플릿이 사용될 수 있는 위치를 분명하게 나타낸다. 데이터 베이스(130)는 또한 템플릿의 버전의 트랙을 유지한다.

도 1의 새로운 코드 정보 데이터 베이스(140)는 코드 생성 시스템에 의해 생성된 코드에 관한 정보를 포함하고 있다. 예컨대, 생성된 코드에 선언된 유저 규정형 타입 및 선언을 포함한 파일명을 포함한다. 이 정보는 인클루드 파일을 생성하기 위해 사용된다.

도 1의 명세 데이터 베이스(120)는 미들웨어 명세내의 구문 및 의미 정보를 코드 생성 엔진(150)에 의해 효율적으로 사용될 수 있는 형태로 인코딩해서 디바이스 미들웨어 용한 코드(180)를 생성한다. 데이터 베이스(120)의 버전은 전형적으로 대응하는 미들웨어 명세의 버전과 매칭된다. 그러나, 유저가 새로운 특성을 지원하도록 데이터 베이스(120)를 수정하면, 데이터 베이스(120)는 원래의 버전 번호의 추가부분으로서 제공된다. 버전 정보는 코드 캐시(155)로부터 적절한 코드를 생성하기 위해 사용된다.

데이터 베이스(120)는 통합 코드(180)에서 특정 타입의 디바이스로 특정된 부분을 생성하는데 필요한 정보를 제공한다. 예컨대, HAVI 실시예에서, 데이터 베이스(120)는 디바이스 제어 모듈(DCM) 및 기능 제어 모듈(FCM)용 통합 코드를 생성하는데 필요한 정보를 포함한다.

도 3 내지 도 7은 각각의 FCM에 대해 유지되는 정보의 실시예를 나타낸다. 실시예에서 정보는 트리로 조직된다.

도 3은 본 발명에 따른 미들웨어 코드 생성 엔진(150)에 의한 사용을 위한, 기능 제어 모듈(FCM) 데이터 베이스(310)의 상위 세가지 레벨의 블록도를 도시하고 있다. 제 2 레벨은 제네럴 FCM 명세 노드(322), VCR(324)과 같은 FCM 타입으로 인덱스된 명세 정보 노드, 및 튜너(326) 등을 포함한다. 제네럴 FCM 명세(322)는 모든 FCM에 대해 적용가능한 정보를 포함하고, 다른 명세는 특정 타입의 디바이스에만 적용가능하다. 제 3 레벨은 제 2 레벨의 패턴트 노드에 의해 유지되는 정보의 타입을 도시한다. 예컨대, 도 3에 도시된 바와 같이 VCR 노드(322)에 대해서, 제 3 레벨은 데이터 구조(331), 애플리케이션 프로그램 인터페이스(API:332), 이벤트(333), 및 중지 속성(334) 노드를 포함한다. 제 3 레벨은 이 FCM 타입(324)이 인터랙트하는 HAVI 매니저(예컨대, 리소스 매니저, 스트림 매니저 등)를 기록하는 노드와의 인터랙트(335)를 포함한다. 모듈 캐시 노드(336)는 유저에 의해 저장된 모듈 코드(284)의 버전의 트래킹을 유지하고, 도트 모듈이 생성되는 조건을 기록해서 요구된 생성 조건과 매칭되는 코드가 검색될 수 있다.

HAVI의 경우에, DCM은 일 이상의 FCM을 포함한다. 본 발명에 따라서, 중지 속성 노드(334)가 FCM 노드(도시 생략)로 대체되어 있다는 것을 제외하면, 루트 노드 DCM(도시 생략)의 구조는 도 3의 FCM(310)과 유사하다. FCM 섹션은 (특정 기능을 포함하는 디바이스에 대응하는)디바이스 제어 모듈에 포함된 기능 제어 모듈(레코더, 플레이어, 튜너와 같은)의 트래킹을 유지한다.

도 4는, 본 발명에 따른 도 1의 미들웨어 코드 생성 엔진에 의해 사용되는, 도 3의 데이터 구조 데이터 베이스(311)의 데이터 구조 엔트리(331')의 블록도이다. 이해를 돕기 위해서, 프라임 심볼(')은 참조번호 뒤에 사용되어 동일한 참조번호를 가진 데이터 아이템의 실시예를 나타낸다. 데이터 구조(331')는 예컨대 VCR 타입과 같은 특정 FCM 타입으로 한정된 데이터 구조를 기록한다. 도 4는, FCM 타입으로 지정된 인클루드 파일 및 선언 문장을 생성하고, 데이터의 유효성을 체크하는 코드(185)를 생성하기 위해 사용되는 각각의 데이터 구조용 정보(411-424)를 나타낸다. 각각의 데이터 구조(331')는 관련 데이터 타입(411) 및 식별명(412)을 가지고 있다. 바람직한 실시예에 있어서, 데이터 구조(331')는 개별적인 특성(418) 및 선택적으로 소정의 적법한 값의 세트(416)를 가지는 것으로 특징지어 진다. 각각의 데이터 구조(331')는 일 이상의 멤버(414)를 포함하고, 각각의 멤버는 주어진 타입(422) 및 식별명(424)을 가지고 있다.

미들웨어가 FCM을 제어할 수 있게 하기 위해, FCM 코드는 서비스를 미들웨어에 보낸다. 예컨대, VCR FCM은 재생, 녹화, 되감기 등의 서비스를 미들웨어에 보낸다. API는 미들웨어 명세에 의해서 모든 서비스를 보내는 것으로 규정되고, 명세 데이터 베이스(120)는 코드 생성을 목적으로 이 API를 기록한다. 본 발명에 따른 도 1의 미들웨어 코드 생성 엔진(150)에 의한 사용을 위해, 도 5는 도 3의 API 블록(332)에 대응하는 서비스 API 엔트리(332')의 블록도를 도시하고 있다. 도 5는 각각의 서비스 API(332')를 위해서 유지되는 예시적인 정보(511-524)를 도시하고 있다. 이 정보는 다양한 모듈의 서비스와 관련된 메소드(기능)를 생성하는데 사용된다. 예컨대, VCR FCM 모듈은 "재생", "녹화", "되감기", "빨리감기" 등의 메소드(기능)를 포함하고 있다.

프로바이더 엔트리(511)는 이 서비스가 서버 또는 클라이언트에 의해 제공될 것인지를 나타낸다.

리턴 타입 엔트리(512)는 서비스(즉, 서비스를 보내는 방법:332')에 의해 리턴된 아이템의 데이터 타입을 도시하고 있다.

네임 엔트리(513)는 예컨대, 재생, 녹화 또는 빨리감기 등과 같은, 서비스(332')의 이름을 기록한다.

인수(argument) 엔트리(514)는 서비스(332')로 전송될 또는 서비스(332')로부터 리턴될 파라미터를 기록한다. 인/아웃 엔트리(552)는 파라미터가 판독 전용(인력), 기록 전용(출력), 또는 기록-판독(입출력 모두)인지를 나타내고, 데이터 구조 엔트리(524)는 각각의 인수의 구조를 나타낸다.

특성 엔트리(515)는, 예컨대 서비스가 개인용 링크를 통해서 물리적인 디바이스와 통신되어야 하는가 및 개인용 링크에 인터페이스 코드(180: 도 1)를 생성하는데 사용되는가와 같은 서비스에 관한 정보를 기록한다.

부작용 엔트리(516)는 그 값이 API(332')의 액추에이션에 의해 변한 경우에 데이터 구조(326)를 나타낸다. 역효과 엔트리(516)는 변화 이후에 각각의 데이터 구조의 새로운 값을 기록한다. 일반적으로, 이들 데이터 구조는 전송된 도 3의 FCM의 데이터 구조 엔트리(331')의 서브셋이다. 역효과(516) 정보는 생성된 코드(180)내에 식별 코드를 생성하기 위해 사용된다.

예측 엔트리(517)는 이 서브와 관련된 예측 및 에러 상태를 나타낸다. 디폴트 코드가 생성되어서 예측 및 에러를 조정할 수 있지만, 전송한 바와 같이, 본 발명의 바람직한 실시예에서 코드 생성 시스템(100)은 유저가 이들 섹션을 수정할 것을 의뢰한다.

유효 호출자 엔트리(518)는 이 API(332')를 호출할 수 있는 유효 소프트웨어 요소 타입을 나타내고, 호출자의 식별번호(identity)의 체킹을 필요로 하는 코드(18)를 생성하는데 사용된다.

메소드 캐시 엔트리(519)는 서비스(332')를 위한 유저 저장형 메소드(기능)의 트래킹을 유지한다. 수정 시간 및 버전 정보에 더해서, 메소드 캐시 엔트리(519)는 메소드(기능)가 속하는 모듈 타입을 기록한다.

도 6은 본 발명에 따른 도 1의 미들웨어 코드 생성 엔진(150)에 의한 사용을 위한 도 3의 이벤트 엔트리(333)에서 이벤트(333')의 블록도를 도시한다. 이벤트 엔트리(333')는 도 3의 FCM 타입(310)이 포함될 이벤트를 기록한다. 도 6은 각각의 이벤트(333')에 유지되는 정보를 나타낸다. 이 정보는 이벤트 관련 메소드(기능) 및 코드 세그먼트를 생성하는데 사용된다. 도 6의 각각의 노드(612, 613, 614, 619, 622, 624)의 의미는 도 5의 각각의 노드(512, 513, 514, 519, 522, 524)의 의미와 같고, 즉 리턴 타입, 네임, 인수, 인/아웃, 데이터 구조 및 메소드 캐시이다.

본 발명에 따른 도 1의 미들웨어 코드 생성 엔진(150)에 의한 사용을 위한, 식별 속성 엔트리(334)의 중지 속성(334')의 블록도를 도시하고 있다. 중지 속성(334')은 속성이 값을 변화시킬 때 통지를 보내도록 FCM을 트리거하는 속성을 기록한다. 데이터 구조(724) 및 인/아웃(722) 엔트리는 엔트리(524, 522)와 관련된 전송한 바와 같이 속성을 규정한다. 도 2의 중지 관련 메소드(기능:282) 및 코드(284) 세그먼트를 생성하기 위해 중지 속성 정보(334)가 사용된다.

도 8은 본 발명에 따른 미들웨어 코드 생성 엔진(150)에 의한 사용을 위한

도 1의 개인용 링크 엔트리(135)에서 개인용 링크(135')의 블록도의 실시예를 도시하고 있다. 개인용 링크 엔트리(135)는 코드 생성 및 특히 각각의 개인용 링크에 관한 정보와 관련된 판매자 특정 정보를 기록한다. 도 8은 링크용으로 유지된 정보의 실시예를 나타낸다. 개인용 링크에 의해 규정된 동작의 이름과 미들웨어 명세에 의해 규정된 이름 사이의 대응관계를 제공해서 미들웨어로부터의 커맨드를 물리 디바이스의 대응 동작으로 변환하는 코드(180)를 생성하는데 사용된다. 링크 상태 응답(823)은 링크로부터 입수되는 응답의 이름을 기록하고, 코드(180)를 생성해서 이 응답을 핸들링하는데 사용된다.

도 9는, 본 발명에 따른 미들웨어 코드 생성 엔진(150)에 의한 사용을 위한 링크 상태 응답 엔트리(823)의 링크 상태 응답(823')의 블록도를 도시하고, 각각의 링크 상태 응답을 위해 유지되는 정보를 나타낸다. 명칭 엔트리(922)는 응답의 명칭, 예컨대 SUCCESS 또는 NOT_IMPLEMENTED이다. 이벤트 트리거형 엔트리(924)는 응답을 핸들링하는 부분의 이벤트를 나타낸다. 각각의 이벤트(924)용으로 유지되는 정보는 이벤트 명칭(931), 트리거링 동작(932), 상태 데이터가 판독되는 서버(933) 즉 타겟 서버, 취해질 액션(판독 또는 전송:934), 및 이벤트가 전송될 경우의 파라미터값(935)을 포함한다. 메소드 캐시 엔트리(926)는 링크 상태 응답(832')의 유저 저장형 메소드(기능)의 트렉을 유지한다.

도 10은 예컨대 간단한 HAVI의 개념을 사용해서, 본 발명에 따른 미들웨어 코드 생성 시스템(100)에서 생성되는 코드(180, 185)의 설치의 블록도(1000)를 나타내고 있다. 1394 스택(1040)과의 IEEE-1394 인터페이스를 통한, 두개의 HAVI 노드(1010, 1020)로의 디바이스 주 기능부(1052)의 제어를 가능하게 하도록 구성된 디바이스(1050)가 도시된다. HAVI 노드(1010, 1020)는 HAVI 소프트웨어 요소(1012), HAVI 메세징 시스템(1030), 및 1394 스택(1040)을 포함하는 소프트웨어 컨테이너이다. 본 실시예에서, 테스트 애플리케이션(185)이 노드(1010)에 제공되고, 디바이스 FCM가 노드(1020)에 제공된다. 본 실시예의 노드(1010)의 애플리케이션(185)은 노드(1010)의 FCM 클라이언트 코드(180a)를 통해서 노드(1010)의 FCM 서버 코드(180b)와 인터랙트한다. 애플리케이션(185)을 대신해서 동작하는 FCM 클라이언트 노드는 HAVI 메세징 시스템 및 1394 스택을 통해서 노드(1020)의 FCM 서버 코드와 통신한다. 이후에 FCM 서버(1020)는 애플리케이션(185)의 제어하에서 디바이스(1050)와 인터랙트한다.

전술한 부분은 본 발명의 원리를 단지 예시화한 것이다. 분명하게 명시되어 있지 않지만, 본 발명의 원리를 실시하고 있고, 첨부된 청구의 범위의 사상과 범주내에 있는 다양한 장치를 당업자가 개발할 수 있을 것이라는 것은 자명할 것이다.

(57) 청구의 범위

청구항 1.

미들웨어 명세에 기초해서 미들웨어 정보를 포함하도록 구성된 명세 데이터 베이스와,

상기 명세 데이터 베이스에 동적으로 연결되고, 상기 미들웨어 정보에 기초해서 미들웨어 통합 코드를 생성하도록 구성된 미들웨어 코드 생성 엔진

을 포함하는

프로그램 생성 시스템.

청구항 2.

제 1 항에 있어서,

상기 프로그램 생성 시스템은 일 이상의 템플릿을 포함하도록 구성된 템플릿 데이터 베이스를 더 포함하고,

상기 미들웨어 코드 생성 엔진은 상기 미들웨어 정보로부터의 데이터를 사용해서 일 이상의 템플릿의 인스턴스 생성(Instantiation)을 통해서 상기 미들웨어 통합 코드를 생성하도록 구성되는

프로그램 생성 시스템.

청구항 3.

제 1 항에 있어서,

상기 프로그램 생성 시스템은 각각의 링크 타입으로 지정된 링크 정보를 포함하도록 구성된 개인용 링크 데이터 베이스를 더 포함하고,

상기 링크 정보는 상기 각각의 링크 타입에 의해 사용되는 링크 동작 명칭과 상기 미들웨어 정보에 포함된 미들웨어 동작 명칭 사이의 매핑을 포함하고,

상기 미들웨어 코드 생성 엔진은 또한 상기 링크 정보에 기초해서 미들웨어 통합 코드를 생성하도록 구성되는

프로그램 생성 시스템.

청구항 4.

제 1 항에 있어서,

상기 미들웨어 코드 생성 엔진은 또한 상기 미들웨어 정보에 기초해서, 샘플 테스트 애플리케이션 코드를 생성하도록 구성되는

프로그램 생성 시스템.

청구항 5.

제 1 항에 있어서,

상기 미들웨어 코드 생성 엔진에 동적으로 연결되고, 연속 재사용을 위해 미들웨어 통합 코드의 적어도 일부를 저장하도록 구성된 코드 캐시를 더 포함하는

프로그램 생성 시스템.

청구항 6.

제 1 항에 있어서,

상기 미들웨어 코드 생성 엔진은 또한 미들웨어 통합 코드의 적어도 일부를 사용자가 수정하는 것을 용이하게 하는 가이드라인 정보를 생성하도록 구성된

프로그램 생성 시스템.

청구항 7.

제 1 항에 있어서,

상기 미들웨어 코드 생성 엔진은

코드 모듈을 생성하도록 구성된 모듈 생성기와,

메소드 코드를 생성하도록 구성된 메소드 생성기와,

코드 문장(code statement)을 생성하도록 구성된 코드 세그먼트 생성기

중 적어도 하나를 포함하는

프로그램 생성 시스템.

청구항 8.

제 7 항에 있어서,

상기 메소드 생성기 및 상기 코드 세그먼트 생성기 중 적어도 하나는 메소드 템플릿 및 코드 세그먼트 템플릿에 각각 기초해서 메소드 코드 및 코드 문장을 생성하도록 구성된

프로그램 생성 시스템.

청구항 9.

제 1 항에 있어서,

상기 미들웨어 정보는 상기 미들웨어 정보에 대응하는 네트워크를 통해서 제어되는 디바이스의 일 이상의 기능에 애플리케이션을 인터페이스시키기 위해 인터페이스 정보를 포함하는

프로그램 생성 시스템.

청구항 10.

제 1 항에 있어서,

영세 데이터 베이스의 제작을 용이하게 하는 것을 포함하는 유저 인터페이스 시스템을 더 포함하는

프로그램 생성 시스템.

청구항 11.

미들웨어 명세에 기초해서, 미들웨어 정보를 포함하도록 구성된 영세 데이터 베이스에 액세스하는 단계와,

상기 미들웨어 정보에 기초해서 미들웨어 통합 코드를 생성하는 단계

를 포함하는

프로그램 생성 방법.

청구항 12.

제 11 항에 있어서,

일 이상의 템플릿을 포함하도록 구성된 템플릿 데이터 베이스에 액세스하는 단계를 더 포함하고,

상기 미들웨어 통합 코드를 생성하는 단계는 일 이상의 템플릿의 인스턴스를 생성하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 13.

제 11 항에 있어서,

각각의 링크 타입으로 지정된 링크 정보를 포함하도록 구성된 개인용 링크 데이터 베이스에 액세스하는 단계를 더 포함하고,

상기 링크 정보는 상기 각각의 링크 타입에 의해 사용되는 링크 동작 명칭과 상기 미들웨어 정보에 포함된 미들웨어 동작 명칭 사이의 매핑을 포함하고,

상기 미들웨어 통합 코드를 생성하는 단계는 링크 정보에 기초해서 미들웨어 통합 코드를 생성하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 14.

제 11 항에 있어서,

상기 미들웨어 정보에 기초해서 샘플 테스트 애플리케이션 코드 생성하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 15.

제 11 항에 있어서,

연속 재사용을 위해 상기 미들웨어 통합 코드의 적어도 일부를 저장하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 16.

제 11 항에 있어서,

상기 미들웨어 통합 코드의 적어도 일부를 사용자가 수정하는 것을 용이하게 하는 가이드라인 정보를 제공하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 17.

제 11 항에 있어서,

상기 미들웨어 통합 코드를 생성하는 단계는

코드 모듈을 생성하는 단계와

메소드 코드를 생성하는 단계와,

코드 문장을 생성하는 단계

를 포함하는

프로그램 생성 방법.

청구항 18.

제 17 항에 있어서,

상기 메소드 코드 및 코드 문장 중 적어도 하나를 생성하는 단계는 대응하는 메소드 템플릿 및 코드 세그먼트 템플릿 각각에 액세스하는 단계를 포함하는

프로그램 생성 방법.

청구항 19.

제 11 항에 있어서,

상기 미들웨어 정보에 대응하는 네트워크를 통해서 제어되는 디바이스의 일 이상의 기능에 애플리케이션을 인터페이스시키기 위해 인터페이스 정보를 제공하는 단계를 더 포함하는

프로그램 생성 방법.

청구항 20.

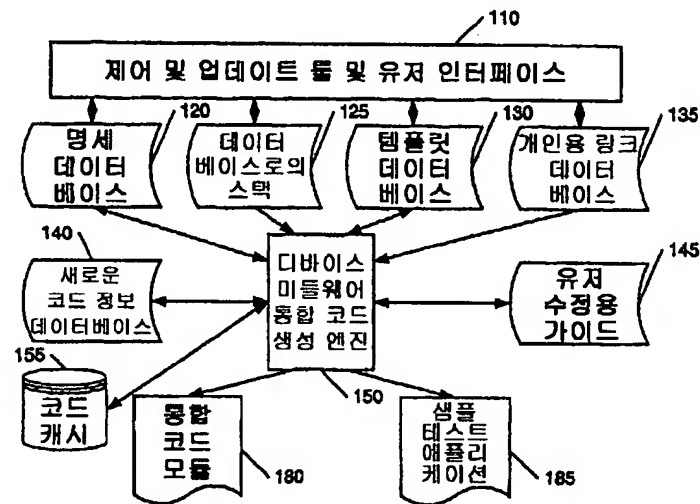
제 11 항에 있어서,

명세 데이터 베이스의 제작을 용이하게 하는 룰을 제공하는 단계를 더 포함하는
프로그램 생성 방법.

요약

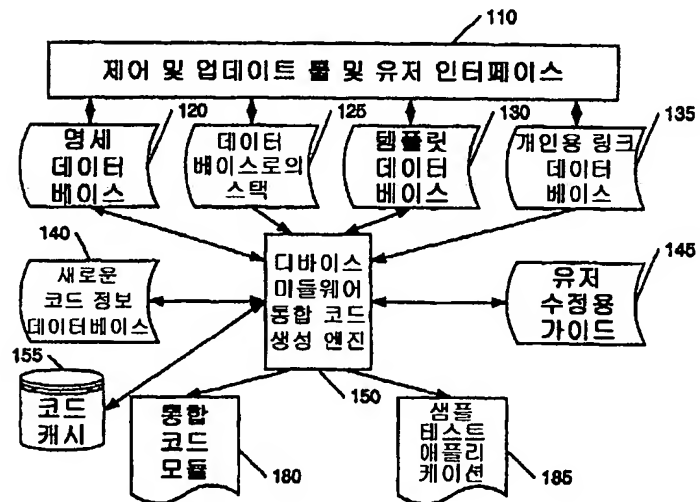
미들웨어 생성 시스템은 디바이스를 네트워크에 통합시키는 프로그램 코드를 생성한다. 다양한 룰 및 유저 인터페이스가 제공되어서 미들웨어 통합 코드를 생성하는데 필요한 정보를 포함한 데이터 베이스를 작성한다. 코드 생성 시스템은 생성된 코드를 유저가 수정하고, 특히 에러 핸들링을 위해 코드를 수정하는 것을 용이하게 하도록 구성된다. 바람직한 실시예에서, 코드 생성 시스템은 애플리케이션의 생성된 통합 코드를 사용하는 방법을 나타내는 것을 돕는 샘플 테스트 애플리케이션을 생성한다.

대표도

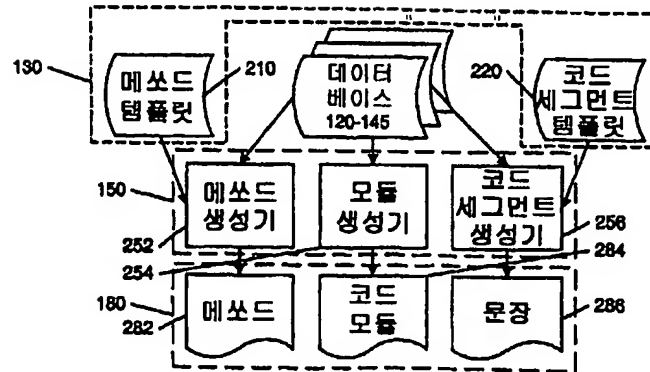


도면

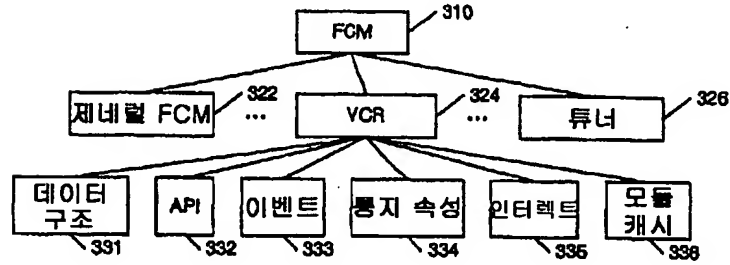
도면 1



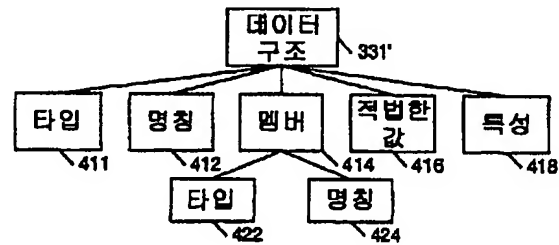
도면 2



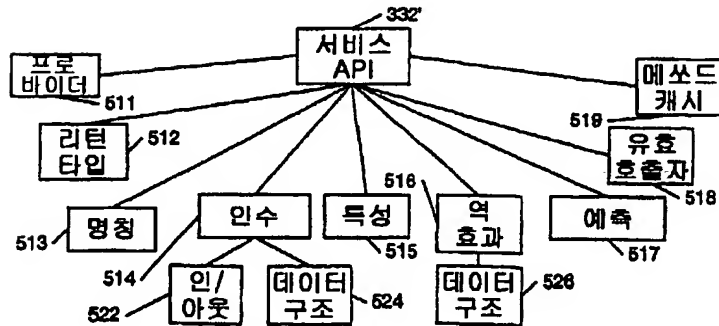
도면 3



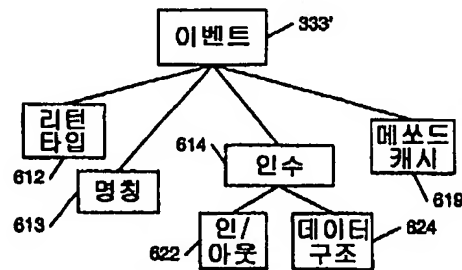
도면 4



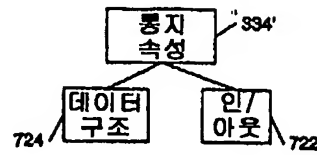
도면 5



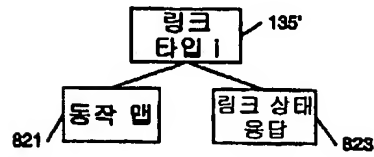
도면 6



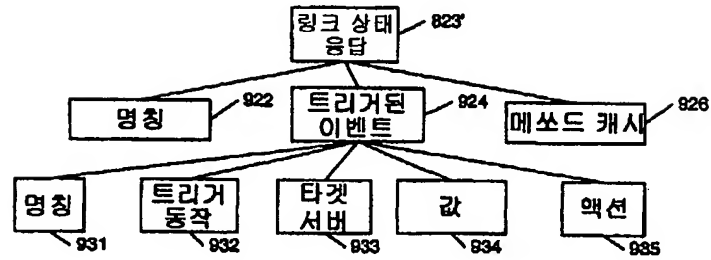
도면 7



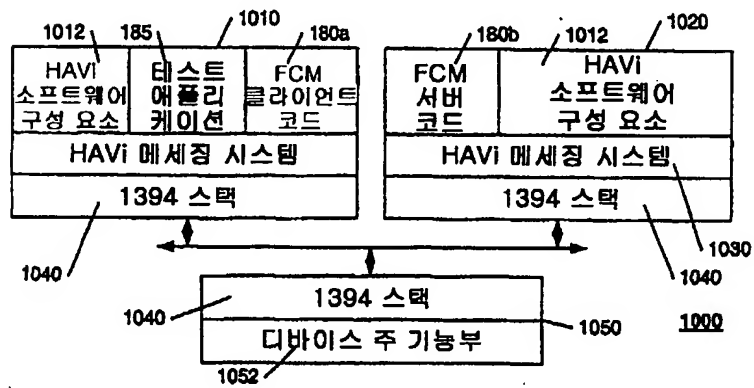
도면 8



도면 9



도면 10



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.